Unicenter

TCPaccess Telnet Server Planning Guide

Version 6.0

Computer Associates

The Software That Manages eBusiness

Booklist

Contents

Chapter 1: Architecture

Chapter 2: Preparing for Installation

Chapter 3: Customizing System Security

Chapter 4: UNIX System Services Support

Chapter 5: User Exits

Appendix A: Editing Tools For Installation

Index

Chapter 1

Architecture

This chapter describes the architecture of Unicenter TCPaccess Telnet Server. It includes the following sections:

- System Architecture—Describes the architecture of the system, including interaction with other systems
- IFS Services—Describes the IFS services in Unicenter TCPaccess Telnet Server, including dump and recovery, messages, SMF, latch, timing, tracing, and operator interface
- Protocol Features—Describes the protocol features of Unicenter TCPaccess Telnet Server

The Unicenter TCPaccess Telnet Server application runs on IBM System/390 and z/OS mainframes running various releases of MVS (see Release Notes). The LE/370 runtime library (SCEERUN) must be either link-listed or included in the STEPLIB. If implementing Server Telnet with SSL, then SGSKLOAD and SCLBDLL must either be link-listed or included in the STEPLIB.

System Architecture

Unicenter TCPaccess Telnet Server runs as an MVS subsystem in its own address space with no operating system modifications. Interfaces to the external system security facility are implemented for signon, data set, and command access security using the MVS SAF router. Interprocess and inter-address space communication is accomplished using cross-memory services, ESA access registers, VTAM, and JES2/JES3. Unicenter TCPaccess Telnet Server is installed and maintained using SMP/E.

Unicenter TCPaccess Telnet Server runs within a runtime environment called the *Infrastructure* (IFS). IFS is a generic, multitasking, runtime environment for MVS system application address space that provides basic services such as cross-memory communications and storage management. A system using the infrastructure is an authorized, operator-started task or job that runs as a subsystem.

IFS Services

IFS Services are described next.

- Dump and Recovery Services include routines to capture dumps of Unicenter TCPaccess Telnet Server and any other involved address spaces (such as an application address space UNIX System Services—MVS) and provide recovery.
- Message Services include routines that write to the operator console or sysout data sets. Messages can be filtered by component and severity, either through product configuration, or an operator command.
- SMF Services includes a standard interface to write records to the SMF data set.
- Timing Services include routines that measure time intervals for various processes.
- Tracing services include routines and macros that keep track of the events that occur within the Unicenter TCPaccess Telnet Server address space. Tracing is done using the IFS internal trace table, the system trace table, through GTF and TCPEEP.
- Operator Interface is a set of various routines that allow significant operational control of the Unicenter TCPaccess Telnet Server address space and the TCP/IP stack.

Protocol Features

The protocol features are:

- Implementation of Server Telnet with support for LU2 and LU0 3270 SNA protocols
- Provision for Telnet Server LU name support (LU security) that associates user ID and terminal access security to an individual Telnet user to use Unicenter TCPaccess Telnet Server in secure environments
- Implementation of Server Telnet TN3270E protocol in accordance with RFC2355 and Functional Extensions
  - Support of LU2 and LU0 3270 SNA protocols
  - Support of LU1 and LU3 SNA protocols
- Implementation of SSL protocol for Server Telnet

Chapter

2

Preparing for Installation

Before you install the Unicenter TCPaccess Telnet Server software, you must prepare your MVS system to accept the product. This chapter provides an overview of the major tasks required to modify the MVS operating system prior to installing Unicenter TCPaccess Telnet Server. These preliminary tasks are described in these sections:

- Determining the MVS Subsystem ID—Describes how to determine the Unicenter TCPaccess Telnet Server MVS subsystem ID
- Customizing the Security Interface—Identifies some considerations for customizing the security interface for RACF, eTrust CA-ACF2, or eTrust CA-TopSecret
- Setting APF Authorization for Common Load Data Sets—Describes the process for authorizing common load data sets for Unicenter TCPaccess Telnet Server

Determining the MVS Subsystem ID

During the initialization process, Unicenter TCPaccess Telnet Server attempts to locate the subsystem control blocks it needs. It looks for the subsystem name in the control blocks. The default name of the subsystem is **ACTN**. You can override the default by changing the SSN=symbolic parameter in the RUNTLN JCL stream.

If you cannot locate the required subsystem control blocks, Unicenter TCPaccess Telnet Server builds them dynamically and places them on the MVS subsystem control block chain. Unicenter TCPaccess Telnet Server does not use the subsystem control blocks if they are in use by another address space.

Dynamic allocation of the subsystem control blocks is recommended. No IPL or maintenance of SYS1.PARMLIB is necessary.

Note: If you prefer to permanently define the subsystem control blocks in your installation, add an entry for the subsystem name in member IEFSSN*xx* in SYS1.PARMLIB. If you do not want to use the default name (ACTN), override the subsystem name on the SSN= parameter in the RUNTLN job stream. You must perform an IPL in order for the change to IEFSSN*xx* to take effect.

Customizing the Security Interface

In installations using external security systems, there may be data access restrictions. The security administrator must ensure that a TCP/IP implementation does not circumvent any restrictions already in place.

If you are using the SSL capable Telnet server, the associated userid of the address space must have superuser privileges.

Setting APF Authorization for Common Load Data Sets

Unicenter Telnet Server LOAD and FTPLOAD data sets require APF authorization. In order to set authorization for these common load data sets, modify the IEAAPF*xx* member of the SYS1.PARMLIB data set.

Modifying SYS1.PARMLIB(IEAAPFxx)

If you do not have a procedure in place for modifying PARMLIB members, use the following steps to update the SYS1.PARMLIB member IEAAPF*xx*:

1. Verify the target name and volume serial of these data sets before proceeding.
2. If you have a procedure in place for modifying PARMLIB members, follow that procedure; if you do not have a procedure in place, proceed to Step 3.
3. Create a full-back member by renaming the current IEAAPF*xx* member and giving it a backup suffix. Copy the renamed member and give it the current suffix. This provides you with a full-back member in the event an error is made during the editing process.
4. Edit the APF authorization member IEAAPF*xx* (or PROG*xx* for ESA Version 4.3 or higher) in SYS1.PARMLIB (where *xx* is the suffix of your member).
5. If using SSL with Telnet, then the GSK.SGSKLOAD data set must be APF authorized. If using the SSL capable server, then CEE.SCEERUN and CBC.SCLBDLL data sets must be APF authorized.
6. You must perform an IPL in order for the changes to take effect.

Note: Some MVS monitoring packages and newer versions of MVS/ESA allow dynamic APF authorization of data sets while the MVS system is running. If you dynamically authorize APF data sets, you must still change the IEAAPF*xx* member or authorization will be lost at the next IPL.

If you are not familiar with changing the IEAAPF*xx* member in SYS1.PARMLIB, consult the *MVS Initialization and Tuning Guide*.

WARNING! *Whenever you make changes to any SYS1.PARMLIB member, be sure you can perform an IPL of your system using an alternate IPL volume or an alternate SYS1.PARMLIB member. Typographical errors can cause catastrophic errors during system initialization, leaving your MVS system in an unusable state.*

Modifying TSO Procedures

Several user interface programs can be executed from TSO address spaces. Some user interface programs have TSO help members that let the users find information on the use and format of each program. Any TSO users who need to reference these TSO help members must have their TSO procedures updated.

Note: Edit any TSO procedures that require access to TSO help members by concatenating a DD statement to SYSHELP. If TRGINDX is specified as T01TCP.V5R0, add the following DD statement to SYSHELP**:**

```
//    DD DSN=T01TCP.V5R2.HELP,DISP=SHR
```

Chapter 3

Customizing System Security

System security is an important consideration in data processing. Products like Access Control Facility 2 (eTrust CA-ACF2), eTrust CA-Top Secret, or Resource Access Control Facility (RACF) help many installations protect valuable data and preserve system integrity.

The following sections describe the security configuration procedures, as required by several security products.

Note: The examples in this chapter use default class and profile names for illustration only; alternate name selection is possible. See the description of the SECURITY statement in the IJTCFG*xx* member for details.

- Security Information in the Log File —Describes a parameter used to display information about the user signon
- Terminal Security Configuration—Describes the parameters that support the security products for the Unicenter TCPaccess Telnet Server terminal security or source security feature
- Types of eTrust CA-ACF2 Security—Describes the security options
- eTrust CA-Top Secret Options—Describes the eTrust CA-Top Secret security options
- RACF Options—Describes the RACF security options

In installations using external security systems, the security administrator usually establishes data access restrictions. The security administrator must ensure that Unicenter TCPaccess Telnet Server does not circumvent these restrictions.

Unicenter TCPaccess Telnet Server interfaces to the MVS security system, via the SAF router, to perform the following functions:

- User ID and password validation

  The user ID and password are validated when sent to Unicenter TCPaccess Telnet Server. Validation occurs at these points:

  - The first time a user tries to use Server Telnet commands that are protected by external security

Security Information in the Log File

Security activity can be monitored by activating appropriate options, either at startup or dynamically via ACTEST. Several categories of security related events can be displayed at execution via messages T00IF070 through T00IF088. Many of these events are frequent occurrences and can quickly flood a log file.

The security categories eligible for monitoring can be initially activated via the XSEC keyword of the SECURITY statement in the IJTCFG*xx* member and can later be enabled or disabled via the ACTEST XSEC command.

The following events are eligible for monitoring:

- ACSECPC—All security calls
- COMMAND—Command authorization calls (for example, ACTEST)
- LOGON—System entry attempts
- LOGOFF—System departures
- ACEE—All ACEE-associated activity

Two other global options are also in effect and are capable of totally disabling either **all** security calls, or just command authorization calls. If you disable security functions at a global level, monitoring cannot be performed. See your system administrator about selective security activation.

For example, you may need to monitor signons, signoffs, and filename accesses for a period. If the startup IJTCFG*xx* SECURITY statement contains XSEC(LOGON LOGOFF), then ACTEST can be executed with XSEC(LOGON LOGOFF OFF) after the monitoring period is over.

Configuring Terminal Security

This section describes the parameters that support the Unicenter TCPaccess Telnet Server terminal security or source security feature.

Terminal Security Configuration

The following parameters of the XSEC parameter on the SECURITY statement in IJTCFG*xx* member are:

- TERMID causes the Unicenter TCPaccess Telnet Server security interface to place a terminal ID into the Terminal field of the signon parameter list for any user attempting a signon to Unicenter TCPaccess Telnet Server. The terminal ID passed during signon attempts is either the remote IP address of the originating host for the user or a VTAM APPL LU name.
- NOTERMID causes the Unicenter TCPaccess Telnet Server security interface to not use the Terminal field in the signon parameter list during signon attempts.

Unicenter TCPaccess Telnet Server defaults to NOTERMID.

Types of eTrust CA-ACF2 Security

Unicenter TCPaccess Telnet Server uses these types of security with eTrust CA-ACF2:

- Signon security—All user ID and password combinations are validated by eTrust CA-ACF2
- Command security—Restricts service to TCPEEP
- Source level security for VTAM LUs and Telnet services

The command security interface restricts access to application segment services. By default, the ACTEST, SYSTAT and TCPEEP are restricted under command security.

To maintain system security, only system programmers and operations personnel should have access to these services.

Because eTrust CA-ACF2 denies all access until permitted, additional steps are required to bring up Unicenter TCPaccess Telnet Server at a site where eTrust CA-ACF2 is installed.

Customizing eTrust CA-ACF2 Version 6 or Later

1. Create a logon ID (LID) record to associate with the startup JCL.

   Follow the installation procedures of your site to create an LID record; make sure these parameters are set in the Privileges Section —Group 2:

   ```
   MUSASS
   NO-INH
   BDT
   ```

   See the *eTrust CA-ACF2 Administrator's Guide* for instructions on creating LIDs.

   Place the LID in the USER field of the startup JCL job card.

   If your site runs eTrust CA-ACF2 6.0 or higher, it is not necessary to set NON-CNCL in the Unicenter TCPaccess Telnet Server LID record.

2. Update GSO records for Unicenter TCPaccess Telnet Server.

3. Protecting packet trace programs from unauthorized use.

   Packet tracing programs must be protected from unauthorized usage. Program T03PTCPE, and its alias TCPEEP, traces packets in and out of the network. Logon IDs, passwords, and perhaps proprietary installation data, can be seen with the packet trace programs.

   Resource rules at the program level are the mechanism within eTrust CA-ACF2 to protect programs. Use the following commands to protect program T03PTCPE and its alias, TCPEEP, in library TELHLQ.LINKLIB where UIDs starting with SYS1 are granted access:

   ```
   ACF

   SET RULE
   COMPILE
   $KEY(TELHLQ)
   $OWNER('Production Telnet Server')
   LINKLIB UID(SYS1-) PGM(T03PTCPE) EXEC(A) READ(A) WRITE(A) ALLOC(A)
   LINKLIB UID(SYS1-) PGM(TCPEEP) EXEC(A) READ(A) WRITE(A) ALLOC(A)
   LINKLIB UID(-) EXEC(A) READ(A)
   END
   STORE
   END
   ```

4. eTrust CA-ACF2: Using the Source Security within Unicenter TCPaccess Telnet Server.

   Unicenter TCPaccess Telnet Server has the ability to pass a source terminal ID to eTrust CA-ACF2 during signon attempts. Unicenter TCPaccess Telnet Server passes either the remote IP address or the actual VTAM terminal ID in the Terminal field during signon attempts.

   Source security customization is an optional feature. Any site that does not currently implement source security can skip this step.

   For more detailed information about source security for terminals, see the *eTrust CA-ACF2 MVS Administrator Guide*.

To use the source security within Unicenter TCPaccess Telnet Server, follow these steps:

Step 1 SAMP member A03ACCES shows the VTAM APPL names starting with A03VLT. This member is a model to use or modify for local use.

The eTrust CA-ACF2 security administrator should group all the VTAM APPL names associated with Unicenter TCPaccess Telnet Server into an X-SGP source record. Currently, there is no mechanism within Unicenter TCPaccess Telnet Server to map VTAM LU usage to specific logon IDs at the VTAM logon points. You do not know which LU will be allocated at these logon points. The LUs used at VTAM logon points within Unicenter TCPaccess Telnet Server are allocated by ACCPOOL. Do not confuse the LUPOOL capability to map IP addresses to logon IDs with ACCPOOL LU customization. All logon ID records that need access to Unicenter TCPaccess Telnet Server through VTAM can then have the new source group added to their source GROUP records.

The eTrust CA-ACF2 security administrator can create X-SGP source records for the A03ACCES SAMP member by issuing these commands for VTAM usage:

```
SET ACF
SET X(SGP)
INSERT A03VLT SOURCE INCLUDE(A03VLT-) ADD
```

The eTrust CA-ACF2 security administrator should check with both the Unicenter TCPaccess Telnet Server and VTAM systems programmers to identify which VTAM LUs are being used by the site for access within Unicenter TCPaccess Telnet Server.

Step 2 All logon IDs that want to sign on to Unicenter TCPaccess Telnet Server must be permitted source authority to the Unicenter TCPaccess Telnet Server IP addresses as specified on the IP address parameter for every NETWORK statement in TCPCFG*xx* member.

A sample NETWORK statement in member TCPCFG*xx* may begin like this:

```
NETWORK IPADDRESS(138.42.224.15)
```

The security system accepts source IDs only in hexadecimal form, so the above IP address must be converted. IP address 138.42.224.15 would use a terminal ID of 8A2AE00F (where 138 = 8A, 42 = 2A, 224 = E0, and 15 = 0F).

Use these commands to create an X-SGP source record at a site to sign on to Unicenter TCPaccess Telnet Server using its default IP address of 138.42.224.15 for source 8A2AE00F:

```
SET ACF

SET X(SGP)
INSERT 8A2AE00F SOURCE INCLUDE(8A2AE00F) ADD
```

All logon ID records that need access to Unicenter TCPaccess Telnet Server must then have the new source entry 8A2AE00F added to their source GROUP records.

Step 3 Any individual logon ID that uses authorized Telnet commands or FTP from a remote site needs READ access authority for the terminal IP address of the remote site. The originating remote IP address is used for all signon attempts.

To create an X-SGP source record at a site to sign on to Unicenter TCPaccess Telnet Server using its host IP address 138.42.224.250 for source 8A2AE0FA, issue the following commands:

```
SET ACF

SET X(SGP)
INSERT 8A2AE0FA SOURCE INCLUDE(8A2AE0FA) ADD
```

This X-SGP source record can now be placed in the source group record for any logon IDs coming in from host 138.42.224.250.

Step 4 An eTrust CA-ACF2 administrator can create a generic X-SGP source record for 8A2AE0- for the local network of 138.42.220 using the following commands:

```
SET ACF

SET X(SGP)
INSERT 8A2AE0 SOURCE INCLUDE(8A2AE0**) ADD
```

You can now place this X-SGP source record in the source group record for any logon IDs coming in from the local network.

Step 5 Activate the X-SGP records using these eTrust CA-ACF2 operator console command:

```
F ACF2,NEWXREF,TYPE(SGP)
```

Step 6 Configure Unicenter TCPaccess Telnet Server to place the terminal ID on all security parameter lists passed to eTrust CA-ACF2 for all signon attempts to Unicenter TCPaccess Telnet Server. If you place TERMID into the XSEC parameter list on the SECURITY statement in the IJTCFGxx member, this happens automatically. By default, terminal IDs are not passed on any signon call.

Step 7 To enable signon checking for Telnet users, add the CPASSWORD option to the Telnet related SERVICE statement(s) in APPCFG*xx* for Telnet ports (typically, 23,1023).

CAUTION! *Activate source security checking only after all eTrust CA-ACF2 customization for Unicenter TCPaccess Telnet Server is completed. eTrust Computer Associates-ACF2 source security can prevent anyone from signing on to MVS, as well as Unicenter TCPaccess Telnet Server, if the customization is performed incorrectly.*

Jobs submitted by TERMID checked logon IDs would fail security unless explicit user IDs and passwords are given when NO-INH is associated with the logon ID of the submitter.

Step 8 To activate the changes in numbers 2 through 8, perform an IPL or issue a GSO console operator refresh. Use this command for the refresh:

```
F ACF2,REFRESH(ALL)
```

eTrust CA-Top Secret Options

This section describes the types of security options available to sites running eTrust CA-Top Secret.

Types of eTrust CA-Top Secret Security

Unicenter TCPaccess Telnet Server uses the following types of security with eTrust CA-Top Secret:

- Signon Security

  All user ID/password combinations are validated by eTrust CA-Top Secret
- Resource eTrust CA-Top SecretSecurity

  Restricts service in the Server Telnet control table
- Source level security for VTAM LUs and Telnet services

The Unicenter TCPaccess Telnet Server command security interface restricts access to services in the Server Telnet control table. ACTEST, SYSSTAT and TCPEEP services should be protected with resource security

To maintain system security, restrict access to system programmers and operations personnel.

eTrust CA-Top Secret Customization

The Unicenter TCPaccess Telnet Server address space functions as a true FACILITY to eTrust CA-Top Secret. Use this setup to enable Unicenter TCPaccess Telnet Server with eTrust CA-Top Secret:

1. Set up a Unicenter TCPaccess Telnet Server FACILITY entry with eTrust CA-Top Secret options as shown in this example:

   ```
   FAC(USERx=NAME=TELHLQ)

   FAC(TELHLQ=PGM=BYP)
   FAC(TELHLQ=ACTIVE,NOABEND,NOASUBM,NOAUDIT,AUTHINIT,ID=c)
   FAC(TELHLQ=NOINSTDATA,KEY=8,LCFCMD,LOCKTIME=0,NOLUMSG,LOG(NONE))
   FAC(TELHLQ=NOMRO,MULTIUSER,NOPSEUDO,NORNDPW,RES,SIGN(M))
   FAC(TELHLQ=SHRPRF,NOSTMSG,TENV,NOTSOC,WARNPW,NOXDEF)
   ```

   In the above example, the Unicenter TCPaccess Telnet Server FACILITY is named TELHLQ. You can use any name up to eight bytes in length. If another name is used, it must be substituted in the setup examples.

   - USER*x* can be any user-defined resource type available at the installation and the *x* value can be any keyboard character.

- For ID=*c*, *c* is a single alphanumeric that represents the FACILITY for reporting purposes (see FACILITY under eTrust CA-Top Secret control options).
- RNDPW (RaNDomPassWords, or return expired new random passwords) can be set on the TCP base product FACILITY. However, only FTP returns all the messages from eTrust CA-Top Secret when the password expires. When RNDPW is placed on your FACILITY definition in an eTrust CA-Top Secret environment, eTrust CA-Top Secret returns a new randomly generated password when an expired password associated with an ACcessor ID (ACID) is correctly presented during signon.

Note: Do not place operands NOPSEUDO, NOMRO, and TENV on the FACILITY definition under eTrust CA-Top Secret 4.3 or above. These operands are no longer supported.

2. Give ACIDs access to the Unicenter TCPaccess Telnet Server FACILITY.

   To permit ACID USER01 access to the Unicenter TCPaccess Telnet Server FACILITY (TELHLQ), the security administrator must issue this command:

   ```
   TSS ADD(USER01) FAC(TELNETA)
   ```

3. Create the Unicenter TCPaccess Telnet Server ACID.

   Build the ACID for the Unicenter TCPaccess Telnet Server address space with the TSS CREATE command.

   The following command creates ACID TELNETA to run as a started task:

   ```
   TSS CREATE(TELNETA) NAME('TCPACCES ACID') FAC(STC) TYPE(USER)
   PASS(NOPW) DEPT(dept_name) MASTFAC(TELHLQ)
   ```

4. Protecting packet trace programs from unauthorized use.

   Packet tracing programs must be protected from unauthorized usage. Program T03PTCPE, and its alias TCPEEP, traces packets in and out of the network. ACIDs, passwords, and perhaps proprietary installation data, can be seen with the packet trace programs.

   Resource rules at the program level are the mechanism within eTrust CA-Top Secret that protects programs. The following commands can be used to protect program T03PTCPE and its alias, TCPEEP, where department SYKSDEPT owns the programs and ACID SYSUSER has access:

   ```
   TSS ADD(SYSDEPT) PROG(T03PTCPE)
   TSS ADD(SYSDEPT) PROG(TCPEEP)
   TSS PER(SYSUSER) PROG(T03PTCPE)
   TSS PER(SYSUSER) PROG(TCPEEP)
   ```

5. Set up Unicenter TCPaccess Telnet Server as a started task.

   If Unicenter TCPaccess Telnet Server runs as a started task, the relationship also must be established in the eTrust CA-Top Secret STC record. The following TSS ADDTO command connects the started task with the ACID defined by the TSS CREATE command.

   Note: This example assumes that the Unicenter TCPaccess Telnet Server PROC name is SWPROC and the ACID defined for use by the Unicenter TCPaccess Telnet Server Task is TELNETA:

   ```
   TSS ADDTO(STC) PROC(SWPROC) ACID(TELNETA)
   ```

6. Set up Unicenter TCPaccess Telnet Server as a batch job.

   If Unicenter TCPaccess Telnet Server is run as a batch job, the relationship is established by the USER= value coded on the job card. In this example, the Unicenter TCPaccess Telnet Server job must be coded with USER=TELNETA.

7. eTrust CA-Top Secret: Using the Terminal Security Class within Unicenter TCPaccess Telnet Server

   Note: Terminal security customization is an optional feature. Any site that currently does not implement terminal security can skip this step.

   Unicenter TCPaccess Telnet Server can pass a terminal ID to eTrust CA-Top Secret during signon attempts. Unicenter TCPaccess Telnet Server passes either the remote IP address or the actual VTAM terminal ID in the Terminal field during signon attempts.

   To use the terminal security class within Unicenter TCPaccess Telnet Server, follow these steps:

   a. Refer to the *eTrust CA-Top Secret Implementation: General Guide* for information about terminal security. Be careful when activating terminal security for the first time. If done incorrectly, no one will be able to sign on to either MVS or Unicenter TCPaccess Telnet Server.

      Research these sample commands for turning on terminal security:

      ```
      TSS LIST(RDT) RESCLASS(TERMINAL)
      ```

      ```
      TSS REPLACE(RDT) ATTR(GENERIC,NODEFPROT) DEFACC(READ)
      ```

      Issuing the following eTrust CA-Top Secret command prevents all undefined terminals from signing on to an address space using terminal security access to your site. This can be very useful in restricting access to Unicenter TCPaccess Telnet Server via IP addresses. Undefined IP addresses are not permitted to sign on to Unicenter TCPaccess Telnet Server.

      ```
      TSS REPLACE(RDT) ATTR(GENERIC,DEFPROT) DEFACC(NONE)
      ```

b. All ACIDs must be permitted READ access authority to the VTAM Terminal APPL names (not ACBNAMEs) that need to access Unicenter TCPaccess Telnet Server through VTAM logon points. There is no current mechanism within Unicenter TCPaccess Telnet Server to map VTAM LU usage to specific ACIDs at the VTAM logon points. You do not know which LU will be allocated at these logon points. The LUs used at VTAM logon points within Unicenter TCPaccess Telnet Server are allocated by ACCPOOL. Do not confuse the LUPOOL capability to map IP addresses to an ACID with ACCPOOL LU customization.

   SAMP member A03ACCES shows VTAM APPL names starting with A03VLT. This member is a model to use or modify for local use.

   The eTrust CA-Top Secret security administrator can use the following commands to define the terminals and designate which ACID can access Unicenter TCPaccess Telnet Server via VTAM utilizing the A03VLT*xx* VTAM APPLs:

   ```
   TSS ADD(acid) TERM(A03VLT)
   TSS PER(acid) TERM(A03VLT) ACCESS(READ)
   ```

   If the eTrust CA-Top Secret system administrator gives access to Unicenter TCPaccess Telnet Server via the VTAM interfaces only to departments SYS1 and ENG, the terminals can be protected as defined in A03ACCES with the following eTrust CA-Top Secret commands:

   ```
   TSS ADD(SYS1) TERM(A03VLT)
   TSS PER(ENG) TERM(A03VLT) ACCESS(READ)
   ```

   The eTrust CA-Top Secret security administrator should check with both the Unicenter TCPaccess Telnet Server and VTAM system's programmers to identify which VTAM LUs are being used by the site for access within Unicenter TCPaccess Telnet Server.

c. All ACIDs wanting to sign on to Unicenter TCPaccess Telnet Server must be permitted READ access authority to the Unicenter TCPaccess Telnet Server IP addresses as specified on the IP address parameter for every NETWORK statement in the TCPCFG*xx* member.

   A sample NETWORK statement in member TCPCFG*xx* may begin like this:

   ```
   NETWORK IPADDRESS(138.42.224.15)
   ```

   The security system accepts terminal IDs only in hexadecimal form, so the above IP address must be converted. IP address 138.42.224.15 uses a terminal ID of 8A2AE00F (where 138 = 8A, 42 = 2A, 224 = E0, and 15 = 0F).

To allow all ACIDs at a site to sign on to Unicenter TCPaccess Telnet Server for a default IP address of 138.42.224.15 with terminal 8A2AE00F, issue these commands:

```
TSS ADD(acid) TERM(8A2AE00F)
TSS PER(ALL) TERM(8A2AE00F) ACCESS(READ)
```

If you replace ACCESS(READ) on the above command with ACCESS(NONE), the eTrust CA-Top Secret security administrator must use the eTrust CA-Top Secret PERMIT command to allow READ access to all ACIDs or departments that need to sign on to Unicenter TCPaccess Telnet Server.

d. Any individual ACID using authorized Telnet commands or FTP from a remote site into Unicenter TCPaccess Telnet Server must have READ access authority for the terminal that represents the IP address of the remote site. The originating remote IP address is used for all signon attempts to Unicenter TCPaccess Telnet Server once a connection to Unicenter TCPaccess Telnet Server is made.

   If local ACID USER01 comes off the local network from host 138.42.224.250, then this ACID needs to be permitted access to this IP address via terminal 8A2AE0FA. This can be done with the following commands:

   ```
   TSS ADD(acid) TERM(8A2AE0FA)
   TSS PER(USER01) TERM(8A2AE0FA) ACCESS(READ)
   ```

   An eTrust CA-Top Secret administrator can allow everyone on his local network (138.42.224) access to Unicenter TCPaccess Telnet Server with the following eTrust CA-Top Secret commands:

   ```
   TSS ADD(acid) TERM(8A2AE0)
   TSS PER(ALL) TERM(8A2AE0) ACCESS(READ)
   ```

e. Configure Unicenter TCPaccess Telnet Server to place the terminal ID on all security parameter lists passed to eTrust CA-Top Secret for all signon attempts to Unicenter TCPaccess Telnet Server. If you place TERMID into the XSEC parameter list on the SECURITY statement in the IJTCFG*xx* member, this happens automatically. By default, Unicenter TCPaccess Telnet Server does not place the terminal ID on any signon call.

f. To enable signon checking for Telnet users, add the CPASSWORD option to the SERVICE statement in the APPCFG*xx* member for Telnet ports (typically, 23,1023).

RACF Options

If a computer site runs RACF, the Unicenter TCPaccess Telnet Server RACF interface automatically becomes active upon installation. This section describes the types of security options available to sites running RACF and how to customize security for Unicenter TCPaccess Telnet Server.

Types of RACF Security

With RACF, the following types of security are active in Unicenter TCPaccess Telnet Server:

- Signon security—All user ID/password combinations are validated by RACF
- Source level security for VTAM LUs and Telnet services

In addition to automatic data set and signon security, Unicenter TCPaccess Telnet Server provides command security. See Customizing Command Security with RACF for instructions on customizing security.

To maintain a high level of system security, only system programmers and operations personnel should have access to these services.

Customizing Command Security with RACF

Unicenter TCPaccess Telnet Server uses the local installation-defined resource classes of RACF to implement command security. Refer to IBM document *SPL: RACF SC28-1343* for additional information regarding the macros and tables described in this section.

1. Modify the installation local class descriptor table.

   Place member ICHERCDE in the SAMP data set in your installation source of ICHRRCDE (local class descriptor table).

   Follow your site's installation procedures to update the local class descriptor table ICHERCDE.

   Member ICHRRCDE in the SAMP data set is an example of the general resource class AC#CMD of Unicenter TCPaccess Telnet Server. The Unicenter TCPaccess Telnet Server general resource class description must be used as shown except for the ID, OPER, and POSIT parameters:

```
AC#CMD ICHERCDE CLASS=AC#CMD,     class name; do not change.
    ID=128,                       Unique ID between 128-255.
    MAXLNTH=8,                    Up to 8 character profile name.
    FIRST=ALPHA,                  First character is alphabetic.
    OPER=YES,                     Allow operations people free reign.
    OTHER=ALPHANUM,
    POSIT=25,                     Must be in range 25-55.
    DFTUACC=NONE                  Must be NONE.
```

   Note: The first four characters of each resource class name must be different from the first four characters of all other class names. One of the four characters should be a national or numeric character to avoid inadvertently choosing a future IBM class name.

2. Modify the local installation-defined router table.

   Member ICHRFRTB in the SAMP data set is an example of the Unicenter TCPaccess Telnet Server router entry. Add the following line from member ICHRFRTB in the SAMP data set into your installation source of ICHRFR01 (local router table) exactly as shown. Do not modify any of its parameters.

```
AC#CMD  ICHRFRTB CLASS=AC#CMD,ACTION=RACF
```

   Follow the installation procedures of your site to update the local router table ICHRFR01.

3. Perform an IPL on the system with a CLPA to activate the local installation router and class descriptor tables.

4. Activate the AC#CMD resource class with the following command:

```
SETROPTS CLASSACT(AC#CMD)
```

5. Follow the installation procedures of your site to create a user ID associated with the Unicenter TCPaccess Telnet Server job or started task.

   This sample ADDUSER command creates user ID TELHLQ associated with Unicenter TCPaccess Telnet Server in group PROD:

   ```
   ADDUSER TELHLQ OWNER(PROD) DFLTGRP(PROD)
                   NAME('TCPACCES ACCESS') DATA('production job')
   ```

6. If your site is running Unicenter TCPaccess Telnet Server as a started task, update member ICHRIN03.

7. Protecting packet trace programs from unauthorized use.

   Packet tracing programs must be protected from unauthorized usage. Program T03PTCPE and its alias, TCPEEP, traces packets in and out of the network. User IDs, passwords, and perhaps proprietary installation data, can be seen with the packet trace programs.

   Program control is a mechanism within RACF to protect programs. The following command turns on program control within RACF:

   ```
   SETROPTS WHEN(PROGRAM)
   ```

   The LINKLIB data set should contain programs T03PTCPE and its alias, TCPEEP. To protect these programs from unauthorized usage in library TELHLQ.LINK on VOLSER SYS001 where user SYSUSER can execute these unauthorized programs, issue the following commands:

   ```
   ADDSD 'TCPACCES.LINK' UACC(EXECUTE)
   RDEFINE PROGRAM T03PTCPE
   ADDMEM('TELHLQ.LINK'/SYS001K/PADCHKK) UACC(NONE)
   RDEFINE PROGRAM TCPEEP
   ADDMEM('TELHLQ.LINK'/SYS001/PADCHK) UACC(NONE)
   PERMIT T03PTCPE ID(SYSUSER) ACCESS(EXECUTE)
   PERMIT TCPEEP ID(SYSUER) ACCESS(EXECUTE)
   SETROPTS WHEN(PROGRAM) REFRESH
   ```

RACF: Using the Terminal Security Class Within Unicenter TCPaccess Telnet Server

Unicenter TCPaccess Telnet Server has the ability to pass a terminal ID to RACF during signon attempts. Unicenter TCPaccess Telnet Server passes either the remote IP address or the actual VTAM terminal ID in the Terminal field during signon attempts.

Terminal security customization is an optional feature. Any site that currently does not implement terminal security may skip this step.

To use the terminal security class within Unicenter TCPaccess Telnet Server, follow these steps:

1. Read the *RACF Security Administrator's Guide* (SC23-3726) for information on terminal security.

   Be very careful when activating terminal security for the first time. If done incorrectly, no one will be able to sign on to either MVS or Unicenter TCPaccess Telnet Server.

   Research these sample commands for turning on terminal security:

   ```
   SETROPTS TERMINAL(READ)
   SETROPTS CLASSACT(TERMINAL) RACLIST(TERMINAL)
   ```

   Issuing the RACF command SETROPTS TERMINAL(NONE) prevents all undefined terminals from signing on to an address space using terminal security. This is useful in restricting access to Unicenter TCPaccess Telnet Server via IP addresses. Undefined IP addresses are not permitted to sign on to Unicenter TCPaccess Telnet Server.

2. All users that need to access Unicenter TCPaccess Telnet Server through VTAM logon points must be permitted READ access authority to the VTAM Terminal APPL names (not ACBNAMEs). Currently, there is no mechanism within Unicenter TCPaccess Telnet Server to map VTAM LU usage to specific user IDs at the VTAM logon points. You do not know which LU will be allocated at these logon points. The LUs used at VTAM logon points within Unicenter TCPaccess Telnet Server are allocated by ACCPOOL. Do not confuse the LUPOOL capability to map IP address to user IDs with ACCPOOL LU customization.

   SAMP member A03ACCES shows VTAM APPL names starting with A03VLT. This member is a model to use or modify for local use.

   The RACF security administrator can use the RDEFINE TERMINAL... and the PERMIT RACF commands to designate which users can access Unicenter TCPaccess Telnet Server via VTAM using the A03VLT*xx* VTAM APPLs.

If the RACF system administrator decides to allow access to Unicenter TCPaccess Telnet Server via the VTAM interfaces only to groups SYS1 and ENG, the terminals can be protected as defined in the SAMP member A03ACCES with the following RACF commands:

```
RDEFINE TERMINAL A03VLT* UACC(NONE)

PERMIT A03VLT* CLASS(TERMINAL) ID(SYS1,ENG) ACCESS(READ)
```

The RACF security administrator should check with both the Unicenter TCPaccess Telnet Server and VTAM systems programmers to identify which VTAM LUs are being used by the site for access within Unicenter TCPaccess Telnet Server.

3. All users who want to sign on to Unicenter TCPaccess Telnet Server must have READ access authority to the Unicenter TCPaccess Telnet Server IP address(es) as specified on the IP address parameter for every NETWORK statement in the TCPCFG*xx* member.

   A sample NETWORK statement in member TCPCFG*xx* may begin like this:

   ```
   NETWORK IPADDRESS(138.42.224.15)
   ```

   The security system accepts terminal IDs only in hexadecimal form, so the above IP address must be converted. IP address 138.42.224.15 would use a terminal ID of 8A2AE00F (where 138 = 8A, 42 = 2A, 224 = E0, and 15 = 0F).

   To allow all users at a site to sign on to Unicenter TCPaccess Telnet Server for a default IP address of 138.42.224.15 with terminal 8A2AE00F, issue the following command:

   ```
   RDEFINE TERMINAL 8A2AE00F UACC(READ)
   ```

   If you replace UACC(READ) on the above command with UACC(NONE), the RACF security administrator must use the RACF PERMIT command to allow READ access to all users or groups that need to sign on to Unicenter TCPaccess Telnet Server.

4. Individual users must be permitted to use their own IP addresses. If local user USER01 comes off the local network from host 138.42.224.250, this user must be permitted access to this IP address via terminal 8A2AE0FA. This can be done with the following commands:

   ```
   RDEFINE TERMINAL 8A2AE0FA UACC(NONE)

   PERMIT 8A2AE0FA CLASS(TERMINAL) ID(USER01) ACCESS(READ)
   ```

   A RACF administrator could allow everyone on his local network (138.42.224) access to Unicenter TCPaccess Telnet Server with the following RACF command:

   ```
   RDEFINE TERMINAL 8A2AE0* UACC(READ)
   ```

5. Configure Unicenter TCPaccess Telnet Server to place the terminal ID on all security parameter lists passed to RACF for all signon attempts to Unicenter TCPaccess Telnet Server. If you use the TERMID option on the XSEC parameter of the SECURITY statement in the IJTCFG*xx* member, this happens automatically. By default, Unicenter TCPaccess Telnet Server does not place the terminal ID on any signon call.

   To activate passing terminal IDs on the security parameter list to RACF for an active Unicenter TCPaccess Telnet Server address space, issue the following command under ACTEST:

   ```
   XSEC TERMID ON
   ```

   You can deactivate passing terminal IDs on the security parameter list to RACF for an active Unicenter TCPaccess Telnet Server address space by issuing the following command under ACTEST:

   ```
   XSEC TERMID OFF
   ```

6. To enable signon checking for Telnet users, add the CPASSWORD option to the Telnet related SERVICE statement(s) in the APPCFG*xx* member for Telnet ports (typically, 23,1023).

WARNING! *Activate terminal security checking only after all RACF customization for Unicenter TCPaccess Telnet Server is completed and the RACLIST profiles have been refreshed (SETROPTS REFRESH TERMINAL). RACF terminal security can prevent signon to MVS, as well as Unicenter TCPaccess Telnet Server, if the customization is performed incorrectly.*

Chapter

4

UNIX System Services Support

This chapter describes configuration information for using Unicenter TCPaccess Telnet Server with IBM UNIX System Services (formerly OpenEdition) for MVS and provides minimal configuration information for Unicenter TCPaccess Telnet Server UNIX System Services Converged socket support. For complete information on configuring and using Unicenter TCPaccess Telnet Server UNIX System Services Converged Socket support, refer to the *Unicenter TCPaccess Telnet Server Communications Server C/Socket Programmer's Reference* and the IBM document *MVS/ESA: Planning Open Edition MVS, BPXB2 MO4/5 SC23-3015-01/02*.

Using UNIX System Services Sockets

The user ID associated with the address space must have Superuser privileges.

Chapter 5

User Exits

This chapter provides information about writing exit routines for Unicenter TCPaccess Telnet Server. It includes these sections:

- The Exits—Defines data areas, macro instructions, and coding conventions and restrictions that apply to all user exit routines in Unicenter TCPaccess Telnet Server

The Unicenter TCPaccess Telnet Server exit facility lets users write exit routines to handle certain specialized requirements within their installation.

Various user exit points are defined within Unicenter TCPaccess Telnet Server to allow the product to be customized. You can configure multiple exit programs for the various exit points. An exit program can communicate with itself across various exit point invocations by establishing an exit context. Exit programs can issue messages, accept or reject various requests, and change or reroute messages. Exit points are defined in message services, at various points in the TCP/IP stack, and in FTP.

The following table lists defined user exit points. These exits are configured in the IJTCFG*xx* member of the PARM data set. See the *Customization Guide* for details on configuring the exit points.

Exit Point	When Invoked	Function
INIT	Startup	Initialize the exit environment
LOG	A message is formatted	The exit can change the message text, reroute the message, or suppress the message
SMFEXIT	An SMF record is about to be written	The exit can reject the writing of the record
TERM	Shutdown	Terminate the exit environment
VTAMBIND	When a BIND RU is received by the TN3270E server	Allow or reject the BIND request

Parameters

Except where noted in the following discussion, parameter lists and the data areas they point to should be left unchanged by the exit program. Changes to other fields are ignored and will not be made effective.

Exit Point ID

Each exit is passed a parameter list pointed to by R1, which includes a fullword identifying the exit point. The first word of the parameter list will be one of the following:

F'0'	INIT exit
F'1'	TERM exit
F'2'	LOG exit
F'21'	VTAMBIND exit
F'22'	SMFEXIT exit

Issuing Messages from Exits

Each exit except the LOG exit is passed the address of a routine that can be invoked in order to write a message to the log or to the operator.

When calling this routine, the following must be provided:

R00	Must contain the value of R13 on entry to the exit program.
R01	Must point to a one-byte message type, followed by 80 bytes of message text. For a description of the message types, see the *Customization Guide*.
R13	Must point to a standard 72-byte register save area.
R14	Must contain the return address.

The message is prefixed by the exit facility with a standard Unicenter TCPaccess Telnet Server message ID (T00EX004) and the exit program name.

Exit Context

At each exit point except INIT, the program is passed a fullword of context. This context word is provided by the program at the INIT exit point, and can be used by the exit program to communicate across exit points.

Return Codes

There are two return codes defined for exits:

- **Return Code 1:** Returned by the exit in R15 when returning to the exit facility. This return code determines whether the next configured exit program is called for this exit point, if multiple exits are configured. This return code should normally be set to zero to allow subsequent exit programs to be invoked.

  A non-zero value causes the exit facility to bypass calling any subsequent exit programs configured at that exit point. (Return code 1 does not apply to the INIT and TERM exits. Register 15 should be set to zero when returning from these exit points.)

- **Return Code 2:** Is part of the parameter list provided on entry to the exit. The exit can set this return code, when appropriate, to reject the current request.

  The value in Return Code 2 after the last exit is called is the value that is returned to the caller of the exit facility. An exit program, therefore, should not change the Return Code 2 value, and, in particular, should not change it from a non-zero to a zero value, unless there is a compelling reason.

Care should be used when setting return codes and when configuring exits. The exit programs are invoked in the same order in which they are configured. A subsequent program can change the Return Code 2 setting from an earlier exit program. An earlier exit program can prevent a subsequent exit program from being called by using Return Code 1.

Exit Work Area

With the exception of the INIT exit point, if requested, the exit facility provides a work area to the exit program upon each invocation. Request the work area with the EWASIZE parameter of the EXIT statement in member IJTCFG*xx*, or with a parameter returned by the program at the INIT exit point (the program parameter takes precedence).

The Exit Work Area (EWA) addressed is passed to the exit program at each exit point as the fifth parameter, following the address of the message-writing routine.

The following entry is added to each exit parameter list table:

+16	04	var	Address of the Exit Work Area (or zero).

The EWA size can be from 1 to 65532 bytes. It is taken from pooled storage. The smallest pool that satisfies the requested size will be used. Pool usage can be monitored via the IFS POOL command.

One of the following pools will be used:

Pool Name	EWA Size
256B	1-252
512B	253-508
01KB	509-1020
04KB	1021-4092
08KB	4093-8188
16KB	8189-65532

Note: These figures are provided to enable the exit program designer to make efficient use of the EWA buffers. It is important that the exit use only the size requested on the EWALENGTH parameter, or by the INIT exit. The exit facility monitors the exit program's use of the EWA, and will force an ABEND if an overrun is detected, *even if space remains in the buffer*.

Exit Recovery Routine

The exit program can supply the address of a recovery routine via a parameter returned at the INIT exit point. The exit facility calls this routine in the event of an ABEND in the exit program. The recovery routine is called in the same mode as the abending exit program, and is passed the System Diagnostic Work Area (SDWA) address and the EWA address (if any). Since the exit's recovery routine is called after the system's Recovery and Termination Manager (RTM) has finished processing the abend, it should perform only local cleanup functions. Any updates to the SDWA are ignored. The recovery routine should **not** attempt to free the SDWA, since this is done by the exit facility.

Recovery Exit

Exit Point: When an abend has occurred in an exit program.

Function: Perform cleanup associated with the exit program.

Dispatchable Unit: Identical with the abending exit program.

Register contents are shown in the following table.

Register	Contents on Entry
R00	SDWA (if processing under an SRB, a copy of the SDWA)
R01	Exit Work Area address (if applicable)
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Note: On return from the recovery exit, R13 must be restored.

Exit Parameter List Mapping Macro—T00DEXPL

A new macro, T00DEXPL, is supplied to map the parameter lists to the exit program at the various exit points. The macro is distributed in the SAMP library.

Using the IEFUSI Sample Exit

When a single Unicenter TCPaccess Telnet Server region is to service many application users, it may require virtual storage beyond the default provided in most installations. Specifying a private area REGION size greater than 16 MB, however, can cause storage allocation problems for system resources below the 16 MB line. In these instances, you may find it necessary to implement a user exit, such as IEFUSI, to ensure that adequate region values are supplied for Unicenter TCPaccess Telnet Server operation. Source for a sample IEFUSI exit is provided in the TCPSAMP distribution data set.

Note: This is a sample only. The region values should be modified to fit your installation's requirements.

The Exits

This section describes the individual exits.

There is a sample EXIT program in the HLQ.SAMP data set.

INIT Exit

Exit Point: Unicenter TCPaccess Telnet Server startup.

Function: The exit is called synchronously at startup, during parsing of the configuration statements in the IJTCFG00 configuration member. You can use it to create an exit context that is passed to the exit program at subsequent exit points. It can also issue messages.

Addressing Mode: 31.

Dispatchable Unit: Task mode (TCB)

Restriction: The exit should not block indefinitely.

Register Contents at Entry:

00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Zero

INIT Exit Parameters Passed.

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'0'.
+04	04	--	Exit context. This word contains zeros on input. The INIT exit can place a word of context in this word. The context is then passed to the exit program at other exit points.
+08	04	--	Return code 2. This should be set to zero.
+12	04	--	Address of the message routine. Note, since the log is not allocated when this exit is called, the message type should be one that will be written by WTO. The LOG exit point can be used to ensure that the message written (T01EX004) is routed to the console.
+16	04	var	Address of the Exit Work Area (or zero).

Offset	Parm Length	Data Length	Description
+20	04	04	Address of a word in which the exit may define which exit points will be driven (except the TERM exit, which is always driven). This word can be built by ORing the exit point ID flag values for the exits that are to be driven. Exit point flags are defined as follows: X'80000000' LOG exit X'40000000' SMFEXIT exit X'00800000' TCPBIND exit X'00400000' SYNRCVD exit X'00200000' SENDSYN exit X'00100000' TCPESTAB exit X'00080000' TCPCLOSE exit X'00040000' UDPBIND exit X'00020000' UDPSEND exit X'00010000' UDPRECV exit X'00008000' RAWSOCK exit X'00004000' RAWSEND exit X'00002000' RAWRECV exit X'00000080' FTPLOGIN exit X'00000040' FTPRSRCE exit X'00000010' VTAMBIND exit For instance, to drive the TCPESTAB, TCPCLOSE, and FTPLOGIN exits, set the value of this word to X'00180080'. **Note:** This request may be overridden by the configuration. See the discussion of the EXIT statement in the *Customization Guide*.
+24	04	Var	Address of the PARM string from the EXIT configuration statement. **Note:** This area is released following the exit point. If the exit program wants to save this string, it must get storage and make a copy.
+28	04	04	Address of a fullword containing the length of the PARM string from the EXIT configuration statement.
+32	04	04	The address of the four-byte Unicenter TCPaccess Telnet Server subsystem ID.

Offset	Parm Length	Data Length	Description
+36	04	04	The address of a fullword area in which the exit program may put the size of the EWA to get.
+40	04	04	The address of a fullword area in which the exit program may put the address of a recovery routine to be called in the event the exit program abends at a subsequent entry point.
+44	04	04	Four-byte product version in C'0v0r' format. For example, Release 5.2 would be x'F0F5F0F2'
+48	04	32	MF subtype mask. Each bit in the mask corresponds to an SMF record subtype 0-255. If the SMF exit point is requested, the exit can turn on those bits corresponding to the SMF record subtypes for which it wants to gain control. The SMF exit point is called for these subtypes, in addition to any SMF subtypes configured on the SMF statement in IJTCFG*xx*. Turning on a bit only causes the exit point to be driven; it does not effect whether the record is written to the SMF data set.

SMF Exit

Exit point: When an SMF record is about to be written.

Function: Allow or suppress writing the record.

Addressing Mode: 31

Dispatchable Unit: Task mode or SRB mode.

Restrictions: Normal restrictions for SRB-mode processing. No SVCs may be issued.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

SMFXIT Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'22'.
+04	04	--	Exit context.
+08	04	--	Return code 2. This should be set to one of the following: 00 - Write the record 04 - Do not write the record.
+12	04	--	Address of the message routine.

Offset	Parm Length	Data Length	Description
+16	04	--	Address of the exit work area
+20	04	08	Address of the SMF record

TERM Exit

Exit Point: Unicenter TCPaccess Telnet Server shutdown.

Function: The exit is called synchronously at shutdown. It can be used to terminate the exit environment and clean up.

Addressing Mode: 31

Dispatchable Unit: Task mode (TCB)

Restrictions: The exit should not block indefinitely.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Zero

TERM Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'1'.
+04	04	--	Exit context.
+08	04	--	Return code 2. This should be left unchanged.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero)

LOG Exit

Exit Point: When a message has been formatted and is ready to be written.

Function – The exit can change the message text, reroute the message, or suppress the message.

Addressing Mode: 31.

Dispatchable Unit: Task mode PC routine or SRB

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

LOG	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

LOG Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'2'.
+04	04	--	Exit context.
+08	04	--	Return code 2. This should be left unchanged. To suppress the message, set the message routing code to zero (see below).
+12	04	--	Zeroes.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	var	Address of the message text. The text may be changed by the exit.
+24	04	04	Address of a word containing the message buffer length. **Note:** This value should not be changed; the message cannot be made any longer than this length. To shorten the message, pad to the right with blanks.
+28	04	04	Address of a word containing a message routing code: X'00000000' Do not issue message X'00000004' Write message to log X'00000008' X'0000000C' Write to log and console. This word may be changed by the exit to change the routing for the message. To suppress the message, set this word to zero.

VTAMBIND Exit

Exit point: When a BIND RU is received by the TN3270E server.

Function: Allow or reject the BIND request.

Addressing Mode: 31.

Dispatchable Unit: SRB mode (the VTAM SCIP exit).

Restrictions: Normal restrictions for SRB-mode processing. No SVCs may be issued.

Register Contents at Entry:

R00	Exit point Id
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

VTAMBIND Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'21'.
+04	04	--	Exit context.
+08	04	--	Return code 2. This should be set to one of the following: 00 – Accept the BIND 04 – Reject the BIND.
+12	04	--	Address of the message routine.
+16	04	08	Address of the local host AF_Inet

Offset	Parm Length	Data Length	Description
+20	04	08	Address of the remote host AF_Inet
+24	04	08	Address of the User ID.
+28	04	04	Address of the TCP session number.
+32	04	08	Address of the SLU name.
+36	04	08	Address of the PLU name.
+40	04	08	Address of the requested application.
+44	04	36	Address of the first 36 bytes of the BIND image (mapped by ISTDBIND).

Appendix

A

Editing Tools For Installation

This chapter describes the edits you must make before installation and how to use the tools necessary for these editions.

It covers the following subjects:

- Setting Up the SMP Environment
- Making Global Changes with the ISPF Editor
- Updating the TCPNAMES ISPF Edit CLIST

Setting Up the SMP Environment

The first time you install Unicenter TCPaccess, you must set up the SMP environment. To do this, you must edit some of the members to allocate common load data sets.

There are two ways to allocate the common data sets:

- Making Global Changes with the ISPF Editor
- Updating the TCPNAMES ISPF Edit Clist

Globally edit the following symbols using values appropriate for your system configuration:

HOLDCL	DSTINDX	DSTUNIT
DSTVOL	LNKINDX	SMPINDX
SMPUNIT	SMPVOL	TLBUNIT
TLBVOL	TRGINDX	TRGUNIT
TRGVOL		
SMPCLAS		SMPINDX2

All Unicenter TCPaccess software products require that you establish a common environment for your installation.

The process includes these tasks:

- Allocating and initializing the SMP and common data sets for Unicenter TCPaccess
- Allocating product-dependent data sets for optional products

Making Global Changes with the ISPF Editor

This is the format of the global change command if you are using the ISPF editor to edit the ALLOCSMP job:

```
c 'old_string' 'new_string' all
```

To change the string SMPINDX to TCPACSS, enter this command:

```
c 'SMPINDX' 'TCPACSS' all
```

To change the string SMPINDX2 to CPT, enter this command:

```
c 'SMPINDX' 'CPT' all
```

If you are changing the HOLDCL symbol to an asterisk in order to use the same SYSOUT class as specified on the JOB statement MSGCLASS parameter, make sure the asterisk is enclosed in single quotes in your global change command, as shown below:

```
c 'HOLDCL' '*' all
```

Important! *Do not modify job ALLOCSMP to use a predefined Consolidated Software Inventory (CSI) in which other software products are installed. That is, Unicenter TCPaccess software products must be installed in their own target and distribution zones.*

Note: Expect return code zero from each step of ALLOCSMP.

Updating the TCPNAMES ISPF Edit CLIST

TCPNAMES is an ISPF Edit CLIST in the CNTL data set. TCPNAMES globally changes all the strings used by the ALLOC*xxx* job streams. TCPNAMES inserts a jobcard and updates the job with your local variables.

The TCPNAMES Clist requires ISPF Version 2 or higher. If the TCPNAMES CLIST is used as distributed, all data sets must have the same high-level qualifier except for the LINK data set, which is prefixed by SYS1. plus the high-level qualifier used for the other data sets.

1. Edit the JOBCARD member in the CNTL data set, changing the JOB statement to match your site's requirements.
2. Copy member TCPNAMES to a fixed length record format CLIST data set that is in your TSO SYSPROC DD concatenation.
3. As you edit the ALLOC*xxx* and other installation members, update the variables in the ALLOC*xxx* job with the parameters passed through this TCPNAMES command:

```
TCPNAMES high_level disk_vol disk_unit tape_vol tape_unit
```

TCPNAMES	Is the command to use to edit the TCPNAMES CLIST.
high_level	Indicates the data set high-level qualifier.
disk_vol	Specifies the disk volume where the data sets are to be created.
disk_unit	Indicates the disk unit type of the volume.
tape_vol	Specifies the volume serial number of the installation tape.
tape_unit	Indicates the address of the tape unit to which the installation tape is assigned.

Usage Notes:

Always run the CLIST TCPNAMES from the command line of the member for every job prior to submitting the job.

If the high-level qualifier you are using is TCPACSS, the disk volume serial number is MVS001, the disk unit type is 3390, the tape volume serial number is TCPACSS, and the tape unit is TAPE, then use this primary line command to update variables:

```
TCPNAMES TCPACSS.V xxx MVS001 3390 TAPE
```

This changes all the strings that need to be changed in the job and replaces the JOB statement with the one in the JOBCARD member.

Note: To run the ALLOC*xxx* jobs, it is not necessary to substitute variables for the *tape_vol* or *tape_unit* parameters. They are used in INSTSMPE.

WARNING! *Whenever you make changes to any SYS1.PARMLIB member, make sure you can perform an IPL on your system using an alternate IPL volume or an alternate SYS1.PARMLIB member. Typographical errors can cause catastrophic errors during system initialization, leaving your MVS system in an unusable state.*

Index

3

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

Computer Associates™